EXHIBIT 11

Web Based Radiology Applications for Clinicians and Radiologists

Eric Feingold, George Grevera, Reuben Mezrich, Steven C. Horii, Satjeet Khalsa, and Le Phan

University of Pennsylvania Medical Center, Philadelphia, PA 19104
Department of Radiology

ABSTRACT

The University of Pennsylvania Radiology Department has developed a suite of Web based applications for clinicians and radiologists to provide wide spread, cost-effective and easy access to radiological information. The Image Viewer application provides clinicians and radiologists access to all diagnostic reports and digital images performed in the last week for all Emergency Dept. (ED), Intensive Care Unit (ICU) and Neuro/CT studies. Image control options including zoom/pan, rotate, flip, and window/level are all available.

The Image Mover/Viewer application gives radiologists and technologists the ability to both move studies between any DICOM Storage Class Provider (SCP) and DICOM Storage Class User (SCU) and to view studies from any DICOM (SCP). All studies viewed with this application automatically transfer through a web server for processing before being displayed.

Web server support requires integration using Perl based CGI scripts with our DICOM/PACS and the MIR/CTN for images and our IDXrad/RIS for reports. Targeted images and reports are automatically routed from the PACS and RIS for storage on the web server. All images sent to the web server are modality specific pre-processed to reduce size and improve contrast. After processing, all images are stored in DICOM and GIF formats. Client support requires web browsers with JavaScript and frame support.

Keywords: Web, DICOM, JavaScript, Viewer

1. INTRODUCTION

The Radiology Department is actively seeking new ways to speed, improve and enhance the delivery of imaging study results to the clinical community. Leveraging existing Web-based technologies is an easy and cost-effective method of providing better access to both the diagnostic report and relevent images that result from a radiological exam. Computers capable of running Web browsers are becoming ubiquitous in our facility as is connectivity to the required network infrastructure. Existing methods of providing images to clinicians has typically involved the installation of network infrastructure, specialized workstations, and either customized (in house) software or expensive vendor based workstations. Although vendors at this year's RSNA were displaying very capable software running on low-cost PC's, the issue of software support and image/report integration loom large as problems for any widespread deployment of clinician targeted radiology workstations. We've chosen to pursue in house development of Web-based applications to solve some of the problems in the department and to hopefully provide better service to the clinical community. Specifically we have focused on the following problems:

1. Access to recent imaging studies and associated diagnostic reports for clinicians.

2. Access to recent imaging studies for radiologists.

3. Movement of imaging studies from image archives to diagnostic workstations that do not support DICOM query/retrieve. Also, there is a need to provide a method of retrieving studies from locations other than the destination workstation.

To address these problems we have deployed a dedicated web server and developed a suite of web-based applications to support image movement, retrieval and viewing. In addition diagnostic report access has been incorporated into the viewer

SPIE Vol. 3035 • 0277-786X/97/$10.00

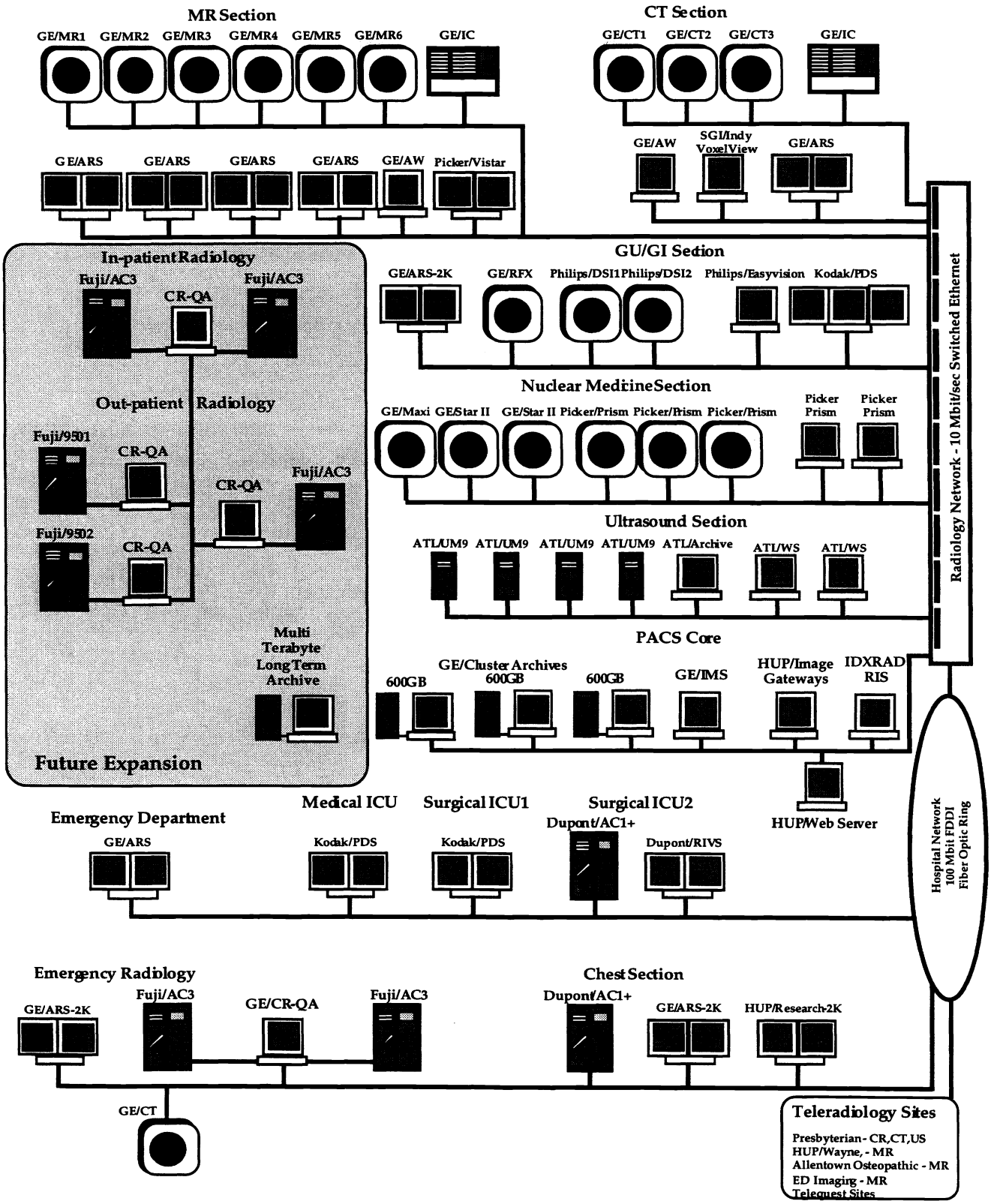

Figure 1. PACS Architecture

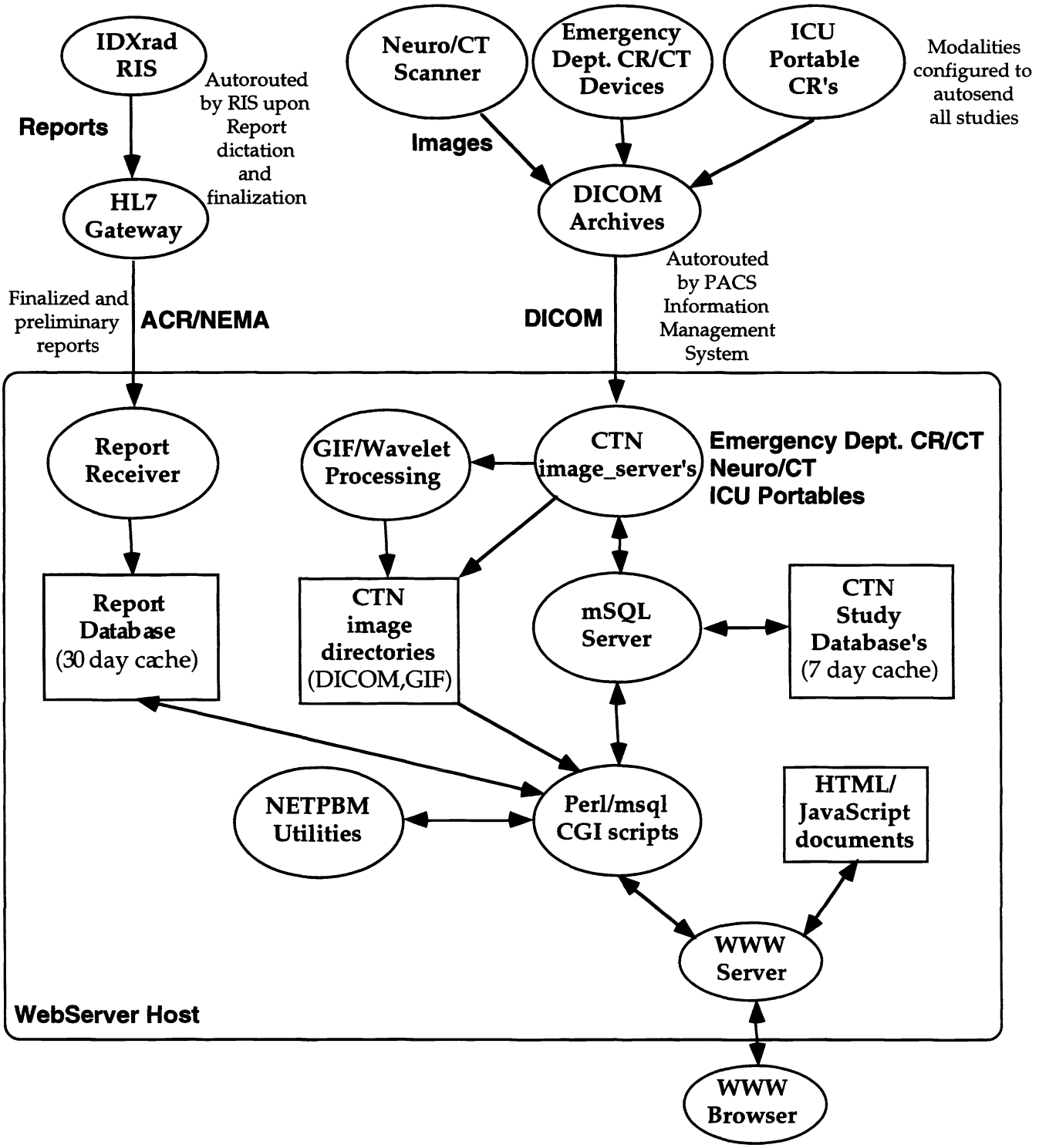

Figure 2. Web Server Host Architecture

applications. Image databases have been set up on the web server to support these applications and configuration changes have been made to our clinical PACS to allow access from and automatic routing to the web server databases. The host server for all applications is a Sun SparcServer 1000E configured with two processors, 128MB RAM, and 10GB of disk running the Solaris 2.5 operating system. The Netscape Inc. Enterprise Server (V2.0) is used as the web server. The Netscape 3.x browser is typically used to run all client applications and has been the test bed for all development.

1.1 PACS Environment

The departmental PACS consists of a switched and routed ethernet environment interconnecting the various imaging modalities, workstations, archives, and information systems (Figure 1). DICOM image output is now available from the following list of acquisition devices:

1. 6 x GE/Genesis MRI scanners
2. 2 x GE/High Speed CT scanners
3. 2 x GE/CTi scanners
4. 2 x GE/Fuji AC3 CR readers via the GE/CR-QA workstation
5. 2 x Sterling/Fuji AC2 CR units via a HUP image gateway
6. 2 x Philips/DSI/RF scanners via the EasyVision/RF workstation
7. 1 x GE/DRX Fluoro scanner

All MR, CT, and AC3/CR images are currently archived across three General Electric cluster archives (600GB on line/archive). An image management system (IMS) receives notification of all archived images and can subsequently automatically route images to DICOM destinations as a function of image header information. Images are automatically routed from the acquisition devices to a designated archive. Upon receipt the IMS sends them to designated workstations and the web server databases.

The radiology information system (RIS) is currently IDXrad V4.x. The RIS is interfaced to the PACS via a custom HL7 interface [1]. Both preliminary and finalized reports are automatically sent to the web server.

2. IMAGE VIEWER

The image viewer application supports access to the latest week of selected departmental studies. Emergency Department CR and CT studies, Neuro/CT studies, and all portable Intensive Care Unit (ICU) chest and abdomen studies are currently supported. Diagnostic reports are available within minutes of transcription. The viewer consists of a DICOM receiver, an image database, a report receiver, a report database, a web server, Perl CGI scripts, and HTML and JavaScript programs (Figure 2). The clinical image management system has been configured to automatically route the selected studies to the web server databases.

2.1 Image Database

We are using the "image_server" process which is included with version 2.8 of the Mallinkrodt Central Test Node (CTN) software [2] to receive and store incoming DICOM images. This process supports DICOM C-STORE messages types for a variety of image modalities including MR, CT, and CR. Image_server is configured to use the Mini SQL (mSQL) database engine [3] for storing all incoming images. Image database access for the viewer is provided by Perl based CGI scripts. We are running Perl v5.002 configured with the MsqlPerl adapter. This adapter allows direct access from Perl to all image databases. One image_server process runs for each class of study input (Emergency CR/CT, Neuro/CT, ICU/CR). The image_server application has been modified to post process images after arrival. Images are modified according to the following rules:

1. CR images (typically 2048 x 1780 x 12 bits) are sub-sampled to 1024 x 890 x 8 bits.
2. CT images are reduced from 512 x 512 x 12 bits to 256 x 256 x 8 bits.

Once modified, additional versions of each image are created to support the viewer application. After size and bit-depth conversion the image is converted to GIF format. To accomplish this we have written a DICOM to Portable GreyMap

(PGM) format conversion utility [4]. We then utilize the NETPBM utility, "ppmtogif" to convert the image to GIF. Finally, a miniature GIF version (50x50 thumbnail) of the image is created using the NETPBM "pnmscale" utility. See Table 1. for the sizes of the various versions of each image. Each night a batch job is run to purge studies older than one week from the image databases.

Original Image	DICOM size	Processed Image	Typical GIF size
2048 x 1780 x 12 bit CR	8 Mbytes	1024 x 890 x 8 bits	700 Kbytes
512 x 512 x 12 bit CT	512 Kbytes	256 x 256 x 8 bits	8 Kbytes

Table 1. Image Sizes

2.2 Report Database

Diagnostic reports from the IDXrad RIS are sent from our PACS Folder Manager [5]. A receiver process runs on the server to receive the reports. Upon receipt, reports are saved in a designated directory using the report accession number as the filename. The latest thirty days of all RIS reports are cached on the server. Each night a batch job is run to purge reports older than one month from the report database.

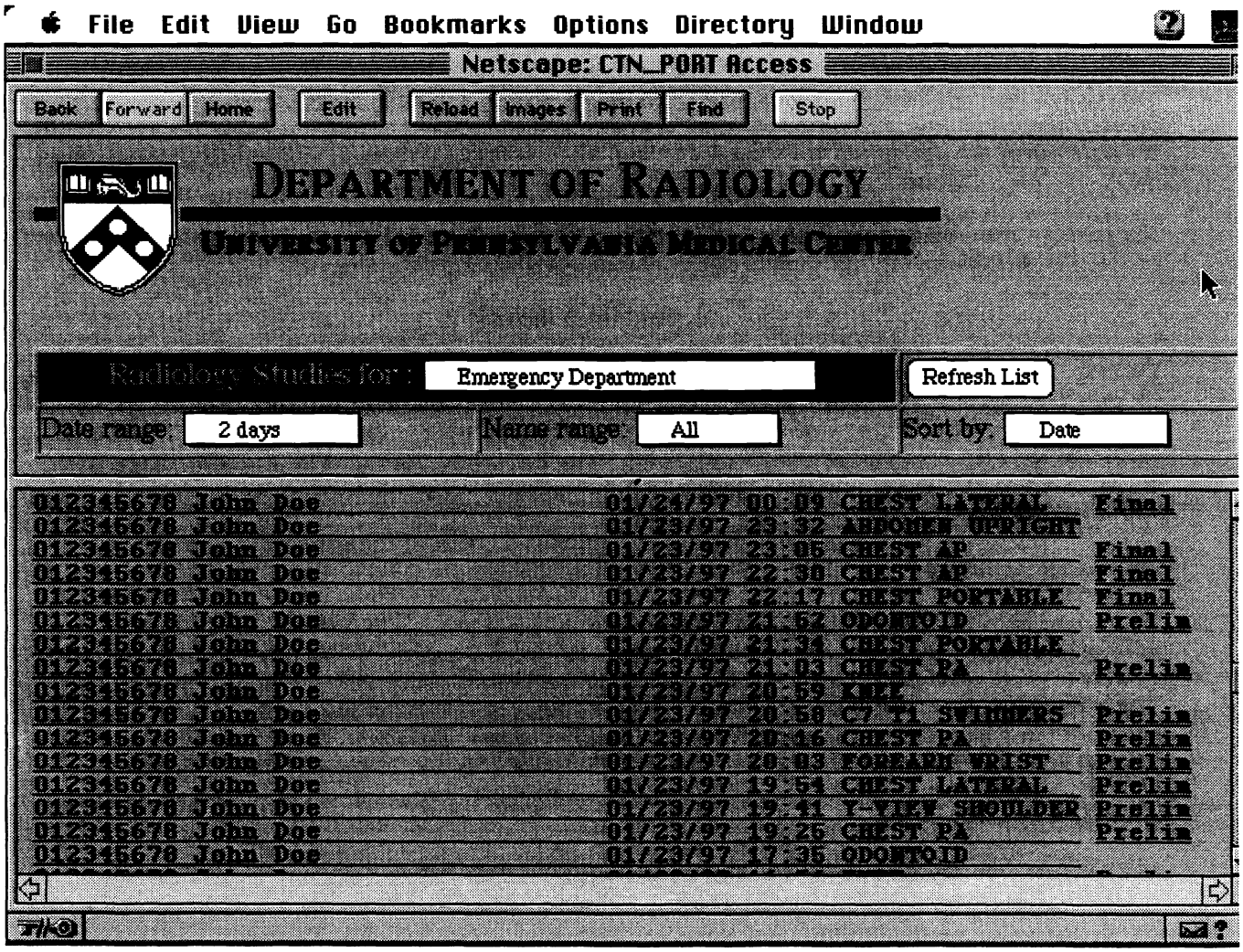

Figure 3. Viewer Study Selection

2.3 Viewer Application

The viewer application consists of a set of HTML, JavaScript, and Perl CGI scripts that all work together. The first screen is a dual-frame HTML display including database query parameter in the top frame and the resultant study list in the bottom frame. Using HTML form selection lists, the user may alter the study query of the selected database. The following fields are selectable from the form:

- Database selection (Emergency Dept., ICU Portables, Neuro/CT)
- Date range (Today, last 2 days, last 3 days, last 4 days, All)
- Patient name (A-E, F-J, K-O, P-T, U-Z)
- Sort by (Name, Date)

Any change in the form immediately invokes a JavaScript function in the form. The function assembles the query parameters into a URL pointing to a Perl CGI that performs the database query which then returns the resulting studies to the bottom frame display. The user can typically find the study of interest within one or two mouse clicks. Each returned study is itself a URL pointing to the image viewing portion of the application. Selecting a given study will load all studies for that patient.

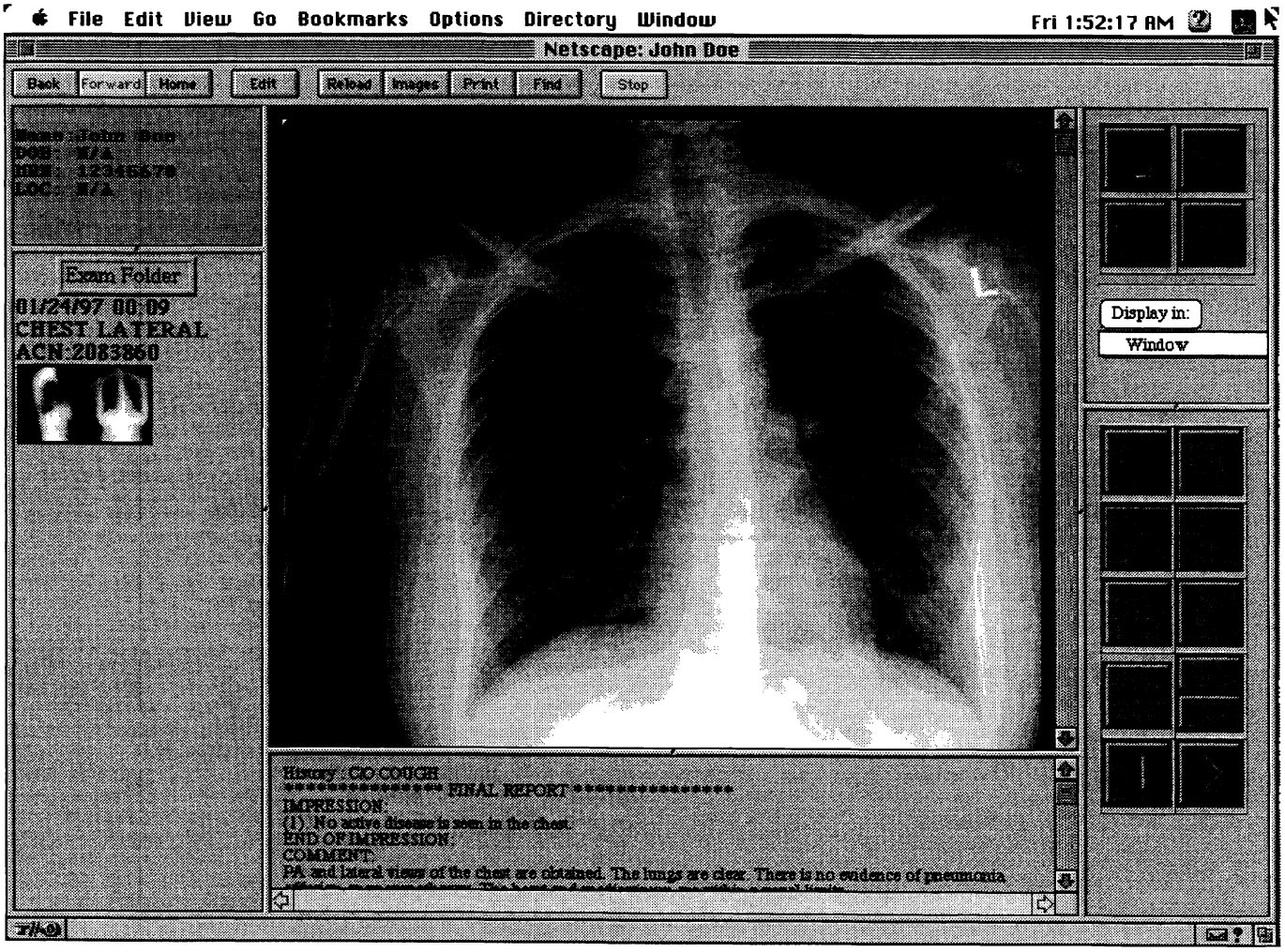

Figure 4. CR Image Viewer

Once a study is selected, a multi-frame HTML document is loaded. The various frames perform the following functions:

1. Demographics Frame: This frame in the top left corner contains the patient name, date of birth, and medical record number.

2. Study Frame: This narrow frame on the left of the screen below the Demographics Frame is loaded in reverse chronological order with thumbnail sized images from each study in the patient's folder. Only the first image in each CT series is displayed. Each study is labelled with the study date and time.

3. Image Frame: This large central frame is for the display of images.

4. Series Frame: This frame appears below the Image Frame when CT series are displayed. It is filled with thumbnails of each image in a selected series.

4. Report Frame: Below the Image or Series Frame is a smaller frame for the display of diagnostic reports.

5. Control Frame: A narrow frame on the right is used for image loading and manipulation buttons.

2.3.1 Image Display

Initially, the demographics and study frames are loaded with all information for the available studies. Selecting a thumbnail image invokes a JavaScript function that loads the full sized image or series into the image frame and the diagnostic report, if available, into the report frame.

Single CR images are loaded and sized to fit vertically in the image frame. The actual size (in pixels) will vary depending on the size of the overall browser window dimensions. A large enough window size could possibly allow full resolution viewing of the selected image. The browser performs any image scaling using pixel replication via the standard HTML tag, <IMG SRC ="image.gif" HEIGHT = 100%>.

CT images are loaded according to the format specification in the control frame (2x2, 3x3, etc.). The images are loaded at full resolution. Scroll bars allow paning over all images if necessary.

2.3.2 Control Functions

Various control functions are provided to allow manipulation of the displayed image. Graphical buttons representing the following operations are displayed in the control frame. Each button invokes a JavaScript function to perform the selected operation. The buttons displayed and their behavior vary depending upon whether a CT or CR study is displayed.

2.3.2.1 CR Controls

A variety of techniques are used to perform manipulation on CR images including including HTML, JavaScript, and CGI scripts. The various controls for CR image manipulation are listed below.

1. Zoom up or down.
2. Rotate right or left.
3. Rotate left or right.
4. Flip vertical or horizontal.
5. Full size.
6. Next or Previous image.

Zooming up or down is implemented using the HTML tag, <IMG SRC ="image.gif" HEIGHT = X%>, where X equals the last zoom factor plus or minus a zoom increment of 25%.

All rotations and flipping functions are performed by Perl CGI scripts that use the NETPBM utilities to process the image. The GIF formatted image is converted to PGM format (giftopnm), rotated or flipped (pnmflip), and converted back to GIF (ppmtogif) for display.

The "Full size" button allows for viewing the current image at full resolution in a new browser window that is sized at the full screen resolution. To accomplish this a JavaScript function opens a new window at full screen size and displays the current CR image at full resolution using the HTML tag, <IMG SRC ="image.gif">. Scroll bars on the window allow paning over the image if the image size exceeds the window size.

The Next and Previous image buttons invoke JavaScript functions which load either the next or previous images in the current study into the image frame.

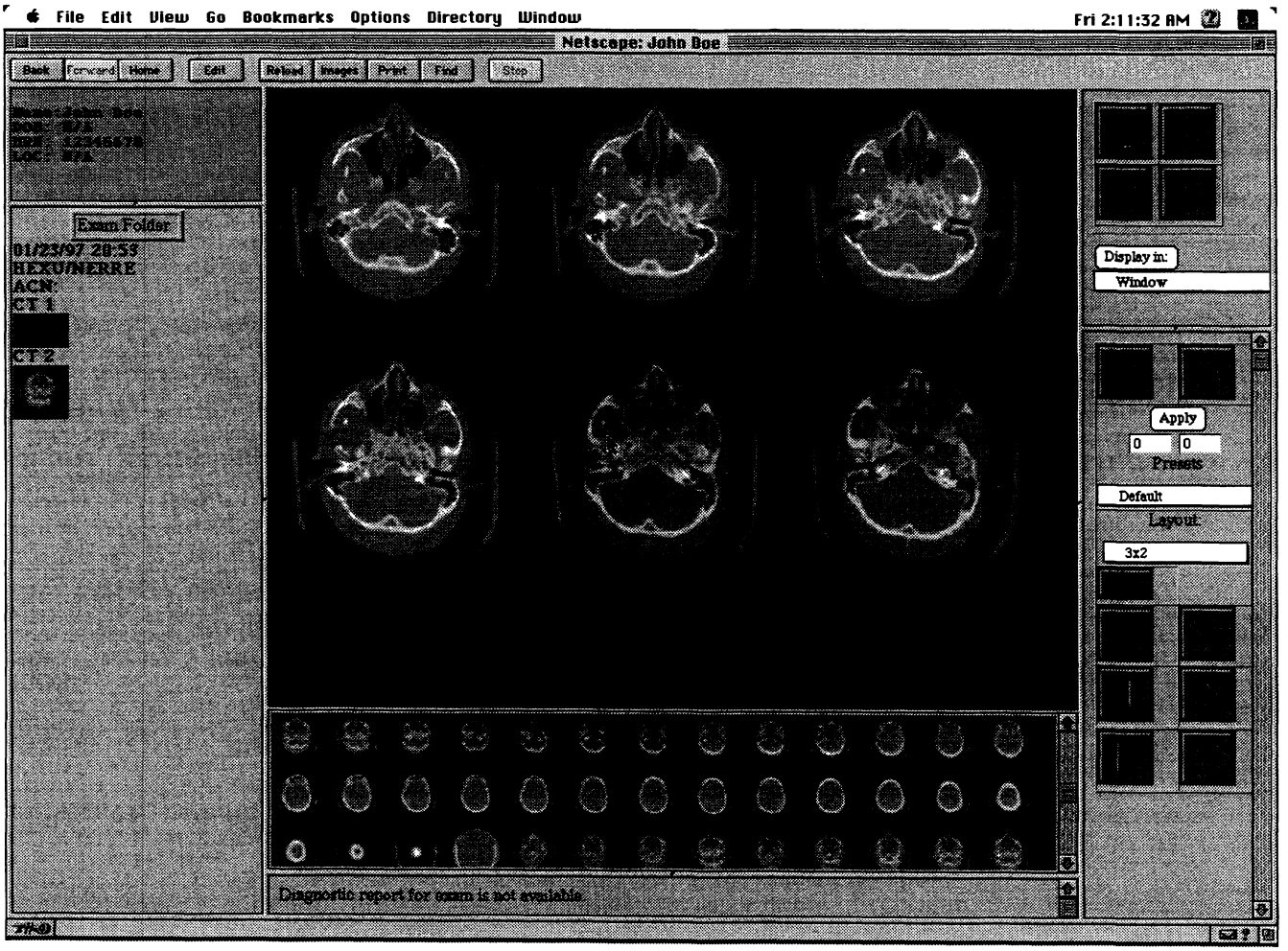

Figure 4. CT Image Viewer

2.3.2.2 Cross-sectional Study Controls

Cross-sectional studies have the following control options:

1. Zoom up or down.
2. Image format.
3. Window/level presets.

4. Cine.
5. Next or Previous image.
6. Next or Previous page.
7. Full size.

Zooming up or down is implemented using the HTML tag, <IMG SRC ="image.gif" HEIGHT = X%>, where X equals the last zoom factor plus or minus a zoom increment of 25%.

Various image display formats varying from 1x1 to 6x6 are available from a pull-down choice list.

Window/level values can either be selected from a pull-down menu or entered numerically. In either case the operation is carried out as a CGI on the web server.

The Next/Previous Image/Page buttons allow navigation throughout the currently loaded series.

The cine option invokes a JavaScript function which loops through each image in the current series. The looping speed and direction is user selectable.

The Full Size button (a simple square) displays a new window and with the entire series loaded in.

2.3.2 Helper Applications

To allow for viewing images outside of the browser we provide the ability to load the image using the browser in a variety of image file formats. Recall that DICOM and GIF versions of each image are stored in the CTN database. Supported formats for loading images into the browser include GIF, DICOM, JPEG, PICT, BMP, and TIFF. The Netscape 3.x browser supports both GIF and JPEG formats natively. Other formats can be supported by either installing "plugins" or helper applications. By configuring the browser appropriately the current image can be loaded into the corresponding helper application or plugin. This feature is invoked by first selecting the desired image format from a pull down menu in the control frame. Pressing the "Display" button then invokes a CGI script outputs the appropriate image file type to the browser. Image file types stored natively (DICOM, GIF) are simply passed directly through to the browser. Other formats are generated by using the NETPBM utilities to convert the native GIF image into the desired format. NETPBM supports virtually every popular image format. We only chose a handful of the available image formats for inclusion in our application.

2.3.3 Viewer Preferences

We have added a set of client specific preference settings to the application to allow individual users to configure various default settings for the viewer. We currently support the following settings:

1. Frame layout: The various frames that make up the viewer can be set to optimize the layout for a given screen. For instance, users may choose to increase the image frame size at the expense of the report frame.

2. Image load format: By default, GIF formatted images are loaded into the image frame. The user may select any of the other image formats to be the default. Thus, whenever an image is to be loaded into the image frame a CGI script will convert (if necessary) the image to the default format before loading it into the browser. For example, if DICOM is chosen as the default format and PAPYRUS (a DICOM image viewer application) is configured as the helper application for DICOM formatted images, all images will be loaded into the PAPYRUS application and NOT be shown in the browser image frame.

Preferences are implemented using Netscape's cookie implementation. This allows a server application to store information in a browser's local preferences file. The preferences may then be retrieved and updated in the future as desired.

2.3.4 Viewer Performance

Formal testing of viewer performance on different computer platforms will be conducted in the future. Table 2 represents a sample of representative times for performing various operations on a common 133Mhz Pentium PC/Windows 95 computer with 48MB RAM running Netscape Navigator V3.0.

Operation	Time (seconds)
single CR image (1Kx1K GIF)	
Load time	7
Flip veritical/horizontal	18
Rotate left/right	17
Full size	3
67 image CT study (256x256 GIF)	
Load all thumbnail images	5
Load all full size images	10
Preset window/level change (all images)	25
Preset window/level change (1 image)	2
Full size	3

Table 2. Perfomance Timings

3. IMAGE MOVER/VIEWER

An Image Mover/Viewer (IMV) application has been developed to support moving studies between two DICOM nodes on the network. As a special case studies can be moved to the browser running the IMV. Specifically, the mover allows one to query and move studies from a DICOM storage class provider (SCP) to any appropriately configured DICOM storage class user (SCU). This application was built on the work presented at in [4].

The first screen of the IMV consists of a three frame screen. The top frame is an HTML/JavaScript form which allows the user to query a selected DICOM destination for studies (Figure 5). Pressing the "Query" button invokes a JavaScript function which bundles all the query parameters into a URL which points to a CGI script which is loaded into the middle frame. The CGI performs a DICOM study level query of the selected SCP and returns all matching studies. Each returned study is itself a link to a CGI with an argument that includes the study UID. Selecting a given study performs a DICOM series level query against the study UID. All series' for the selected study are display in an HTML select list in the bottom frame. The user may then select the desired series to be moved along with a DICOM destination and finally push the "move" button. The move button invokes a JavaScript function which collects the series UID's to be moved. For each series UID a small status window is created and a CGI to perform the actual DICOM move request is invoked. The status window displays the total number of images to be moved and updates the count of moved images as the moves occur. To perform the update the CGI returns single line status information for each image moved. The JavaScript function reads the output of the CGI and in turn updates the status fields in the status window.

A special destination available in the mover is called "Viewer". Images sent to this destination are actually moved to a CTN database as described in the Image Viewer application. Once the selected series have been moved the Image Viewer application is invoked with the desired series automatically loaded.

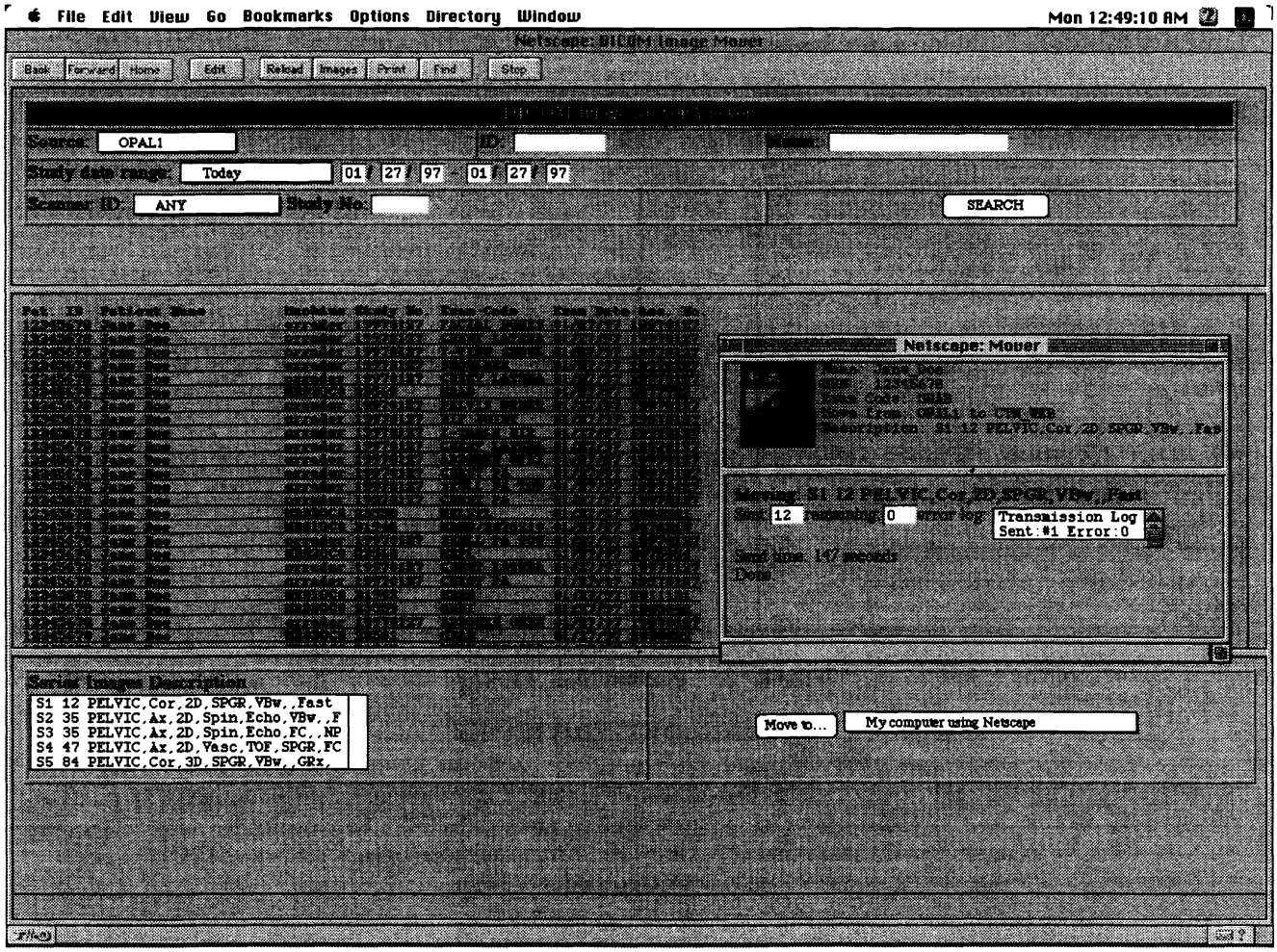

Figure 5. DICOM Image Viewer

4. CONCLUSION

The Image Mover and Viewer applications are now available to all Radiology staff and selected clinical staff. The Image Mover has proved very useful by allowing staff to move studies around the department from their desktop computer. The Viewer application is lightly used today but usage is expected to increase as additional study databases are added to the web server. Ultimately all digital studies will be available from the viewer. One of the largest problems with using the web-based applications is robustness. The Netscape browser often crashes after running the Viewer continuously. It appears that large images stress the capabilities of most browsers. This behavior will most likely improve as the web browser vendors improve there products. We feel confident that the Viewer will ultimately become a widely used method of distributing study results to referring physicians. It is unclear today when and if the Viewer application will replace dedicated DICOM based dual-monitor workstations currently installed in our ICU's.

We are actively improving these applications and trying various mechanisms to allow functionality similar to full featured workstations. We have avoided the temptation to use "plug-ins" to bypass the limitations of HTML and JavaScript. Plug-ins are a step back towards platform specific software and require additional support and configuration for the user. We have actively pursued Java for performing image manipulation functions however today's browsers perform very poorly when handling Java applets that load large images. We expect this to improve in future browser versions and will migrate parts of our applications to utilize Java over time.

One of the greatest advantages of the web-based approach is the ability to leverage existing computing and network infrastructure throughout the institution thereby allowing easy and rapid deployment, upgrading, and support of our applications. This alone outweighs the short-term limitations of today's browsers.

ACKNOWLEDGMENTS

This work was partially supported by Grant P01-CA53141 from the National Cancer Institute, NIH, USPHS, DHHS.

REFERENCES

1. S. B. Seshadri, S. Kishore, and R. L. Arenson, "Software Suite for Image Archiving and Retrieval", *Radiographics* 12, pp. 357-363, 1992.

2. S. M. Moore, A. S. Gokhale, and D. E. Beecher, User's Guide for CTN Demonstration Applications, RSNA, Oak Brook, IL, 1995.

3. Hughes Technologies Pty. Ltd., "Mini SQL - A Lightweight Database Engine", Jan. 1996.

4. G. J. Grevera, E. Feingold, and S. C. Horii, "A WWW to DICOM interface", SPIE Proceedings 2711, pp. 109-117, 1996.

5. S. B. Seshadri, S. Kishore, S. Khalsa, and R. L. Arenson, "Functional design of a folder manager on an image archive.", *Proc. of Computer Applications to Assist Radiology*, R. L. Arenson, R. M. Friedenberg, eds., pp. 242-248., 1990.